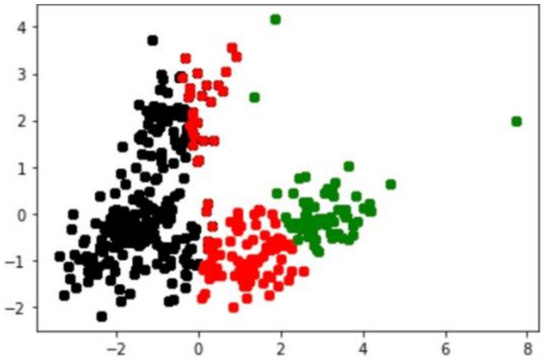
Project1 Report

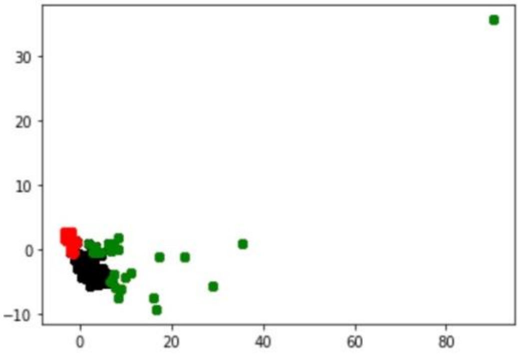
Hayden Trautmann

I began with loading in both the Cho and Iyer datasets as data frames using pandas. I downloaded them into my local directory I was using Jupyter Notebooks in. For data cleaning, I removed outliers which were designated as -1 values in the ground_truth attributes for cho and iyer. Next, I checked to see if there were duplicates in both datasets, which I found none existing. Next, I dropped the gene_id column from the dataset because I didn't want the id's to interfere with my clustering algorithms. Next, I performed PCA Reduction on the datasets. So that I could later graph them on a scatter plot, I reduced both datasets to two dimensions. I tested the K-Means algorithm several times by printing the generated centroids and also printing the elements in their clusters to ensure the elements from each dataset were assigned a cluster correctly and that these clusters were being recomputed based on the mean of the values in the cluster.

My results from the k-means clustering are shown below where I used k = 3 clusters. I was surprised to see the outlier in Iyer because I had removed all outliers identified by their ground truth value as -1.



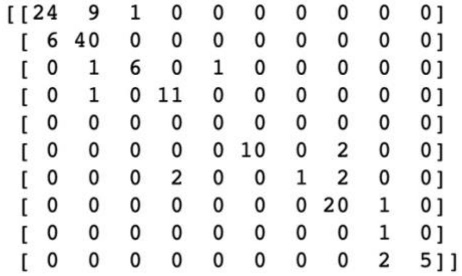
Cho Dataset



Iyer Dataset

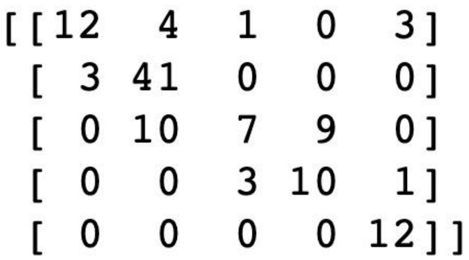
I had an extremely difficult time generating the scatter plots for my K-Means algorithm. I initially used a dictionary to store an array of the elements as the value where the key was the cluster they were assigned to, however, accessing this dictionary and assigning colors to these keys generated many errors. The scatter plot was only generating two of my colors even though I had created 3 that corresponded to my three clusters. It turned out to be indexing issues at the time of plotting. I then tried to add another column to a data frame, which ended up being difficult for me. In the end I used a list of lists which I could index to by converting the given clusters into a tuple and all clusters were accessed.

My implementation of spectral clustering involved four steps. First I represented the data points as a symmetric similarity graph. Doing so I chose to use the k-NN algorithm using the KNeighborsClassifier().



```
[[24  9  1  0  0  0  0  0  0  0]
 [ 6 40  0  0  0  0  0  0  0  0]
 [ 0  1  6  0  1  0  0  0  0  0]
 [ 0  1  0 11  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0 10  0  2  0  0]
 [ 0  0  0  2  0  0  1  2  0  0]
 [ 0  0  0  0  0  0  0 20  1  0]
 [ 0  0  0  0  0  0  0  0  1  0]
 [ 0  0  0  0  0  0  0  0  2  5]]
```

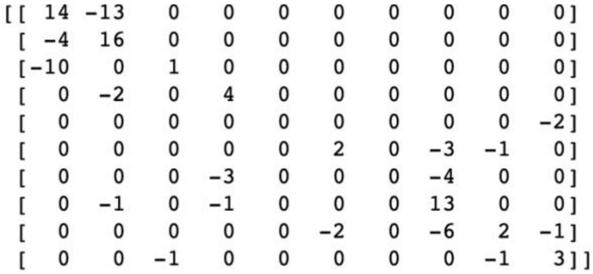
Iyer Symmetric Similarity Graph



```
[[12  4  1  0  3]
 [ 3 41  0  0  0]
 [ 0 10  7  9  0]
 [ 0  0  3 10  1]
 [ 0  0  0  0 12]]
```

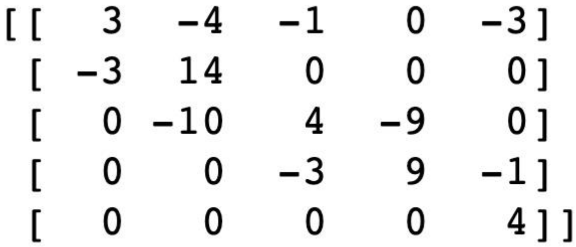
Cho Symmetric Similarity Graph

Next, I compute the graph Laplacian for both datasets using np.linalg.eig().



```
[[ 14 -13   0   0   0   0   0   0   0   0]
 [ -4  16   0   0   0   0   0   0   0   0]
 [-10   0   1   0   0   0   0   0   0   0]
 [  0  -2   0   4   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0  -2]
 [  0   0   0   0   0   2   0  -3  -1   0]
 [  0   0   0  -3   0   0   0  -4   0   0]
 [  0  -1   0  -1   0   0   0  13   0   0]
 [  0   0   0   0   0  -2   0  -6   2  -1]
 [  0   0  -1   0   0   0   0   0  -1   3]]
```

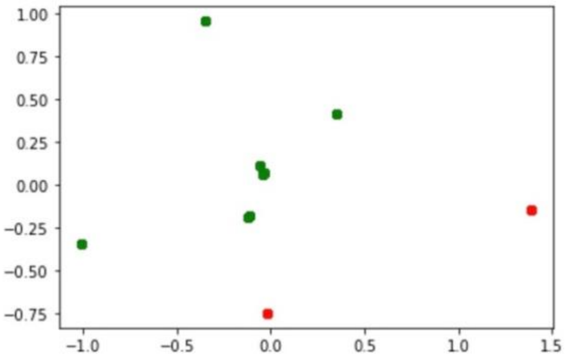
Iyer Graph Laplacian



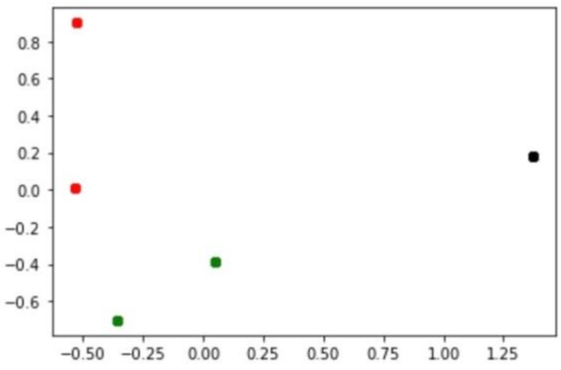
```
[[   3   -4   -1    0   -3]
 [  -3   14    0    0    0]
 [   0  -10    4   -9    0]
 [   0    0   -3    9   -1]
 [   0    0    0    0    4]]
```

Cho Graph Laplacian

After this step, I computed K eigenvectors corresponding to the k smallest non-zero eigenvalues of L. This step generated a eigenvector matrix for both datasets and passed these eigenvectors into my k-means algorithm. I was surprised to see how few data points were returned on the scatter plot. I tried many different methods of computing the eigenvectors but could not seem to get my k-means algorithm working properly on the values. After troubleshooting for hours, the best result I could generate with my spectral clustering implementation is shown below.



Iyer Dataset



Cho Dataset

My results showed that the K-Means algorithm was much simpler to implement than the Spectral clustering algorithm. It returned what appeared to be the more reliable cluster. I found that PCA reduction was an extremely useful task in plotting my results. I conclude that the Iyer dataset showed better clustering than the Cho dataset with both algorithms used. While the Iyer dataset did have one outlier, even after I removed the outliers, it still seemed to have closer and less noisy clusters than the Cho dataset had after performing the algorithms on the preprocessed data.

After doing this project I now have a great appreciation for the K-Means and Spectral Clustering packages from Sci-Kit Learn. I had hardly used Python before this class and I found that using it for machine learning was different than any other coding I had done in the past. Iterating over multi-dimensional vectors of great lengths proved to be a different kind of challenge I had not seen in the past. I really liked using Jupyter Notebooks and found how easy it is to debug and run certain blocks of code. It was much different than most debugging I had done in the past with IDE like Eclipse or VSCode. I look forward to learning more about Python and its machine learning capabilities throughout this semester.